

# Erstellen komplexer elektronischer Dokumente mit SchemaText<sup>1</sup>

Markus Nickl  
Erlangen

Obwohl in der Technischen Dokumentation immer wieder die Forderung nach mehr linguistischem Engagement erhoben wird, ist vielfach noch unklar, wie wissenschaftliche Erkenntnisse in die Praxis umgesetzt werden können. Anhand eines Beispiels soll hier gezeigt werden, daß linguistisches Know-How Eingang in die Entwicklung eines konkreten Produkts finden kann und wie dieses Know-How zu einer verbesserten Funktionalität beiträgt.

## 1 Ausgangslage

Der Ruf nach Hilfe aus der Wissenschaft und insbesondere nach Hilfe aus der Linguistik ertönt wohl in kaum einem Praxisbereich so stark, wie in der Technischen Dokumentation. Gleichzeitig darf die Kenntnis linguistischer Theorien, Modelle und sogar grundlegender sprachwissenschaftlicher Konzepte jedoch beim Gros der Technischen Redakteure<sup>2</sup> nicht vorausgesetzt werden.

Diese Situation führt häufig dazu, daß linguistische Beratungsansätze entweder durch ihre zu hohe Komplexität dem ratsuchenden Technischen Redakteur nicht zugänglich sind, oder daß sich im anderen Extrem die linguistischen Ratschläge auf Alltagsweisheiten und Banalitäten beschränken müssen und somit zu einem Vertrauensverlust in die Hilfsmöglichkeiten der Wissenschaft beitragen.

Eine dritte Möglichkeit, Technischen Redakteuren auf einfache und zeitsparende Weise linguistisches Know-How verfügbar zu machen, wird bisher nur zaghaft genutzt. Sie besteht darin, Fachwissen in die Konzeptionierung von Anwendungssoftware einfließen zu lassen (Knorr 1997, 85).

---

1 Für die Überlassung des Programms möchte ich mich an dieser Stelle bei der Schema GmbH bedanken. Über diesen Aufsatz hinausgehende Informationen zu dem Programm findet man auf ihrer Homepage (<http://schema.de/>, zuletzt eingesehen am 15.10.1998).

Für wichtige Anregungen zu diesem Aufsatz danke ich den Herausgebern und Herausgeberinnen dieses Bandes sowie Herrn Rolf Assfalg.

2 Als Technische Redakteure bezeichnet man die Autoren von Gebrauchsanleitungen und anderen Produktinformationen. Eine Beschreibung der Berufsfeldes findet sich unter <http://www.tekom.de>, zuletzt eingesehen am 15.10.1998.

Im folgenden soll anhand eines solchen Produkts, *SchemaText*, demonstriert werden, wie durch die Berücksichtigung linguistischen Fachwissens ein Werkzeug entstanden ist, das dem Technischen Redakteur die tägliche Arbeit erleichtert, ohne ihn mit einem (im Vergleich zu anderen Programmen) höheren Lernaufwand zu belasten.

## **2 Aktuelle Textproduktionsprobleme in der Technischen Dokumentation**

Um darzustellen, wie die Arbeit des Technischen Redakteurs in der Konzeptions- und Weiterentwicklungsphase mit der Berücksichtigung linguistischen Know-Hows vereinfacht werden kann, ist es zunächst nötig, die Probleme kurz darzustellen, denen die Autoren in ihrer täglichen Arbeit begegnen.

Ein Großteil der Bemühungen in der Technischen Dokumentation dreht sich um den Bereich der Normierung.<sup>3</sup> Neben dem Versuch, insbesondere in größeren Unternehmen, ein einheitliches Erscheinungsbild der Dokumentation zu erreichen, wird die Normierung auch als Grundlage der Qualitätsoptimierung und -sicherung verstanden. Zudem erhofft man sich durch die Normierung der anfallenden Dokumentation eine Basis für die Modularisierung der Dokumente zu schaffen. Dabei wird eine Wiederverwendbarkeit der einzelnen Textteile angestrebt.

Ein weiteres Problem stellt die Übersetzungsunterstützung dar. Produkte werden in zunehmendem Maße für den internationalen Markt produziert. Gleichzeitig wird es durch die nationale Gesetzgebung in fast allen Ländern Pflicht, Gebrauchsanleitungen in der/den Nationalsprache(n) zu veröffentlichen. Die Übersetzung der Anleitungen wird dadurch zu einem entscheidenden Kosten- und v. a. auch Zeitfaktor.

Neben den unterschiedlichen Sprachversionen werden zudem zunehmend häufiger unterschiedliche mediale Versionen (Cross-Media-Publishing) und Repräsentationsformen gefordert (z. B. HTML-Dokumente mit und ohne Frames, Bilder mit geringen Dateigrößen oder mit hochauflösenden Grafiken).

Des weiteren sucht man in der Praxis Lösungen, um die Produktion großer Textmengen zu unterstützen (*Authoring in the large*). Dabei geht es nicht nur um die Frage, wie der einzelne Autor unterstützt werden kann, sondern auch darum, wie man Abstimmungsprozesse beim verteilten Schreiben optimieren kann.

---

<sup>3</sup> Ausführlichere Informationen zur Rolle der Norm in der Technischen Dokumentation bietet Nickl (1997b).

Das letzte Problem, das hier erwähnt werden soll, ist die Frage der Archivierbarkeit der Dokumente. Aus Gründen der Lagerökonomie ist man bereits seit langem dazu übergegangen, Dokumente und Dokumentversionen nur noch elektronisch zu archivieren. Problematisch ist dabei, daß zum einen Speichermedien veralten und zum anderen auch bei den Dateiformaten und Programmen nicht immer eine Abwärtskompatibilität gegeben ist. In der Technischen Dokumentation wird deshalb immer wieder gefordert möglichst weit verbreitete Dateiformate und möglichst „änderungsresistente“ Programme zu verwenden.

### **3 Vorstellung des Programms *SchemaText***

#### **3.1 Charakteristik SchemaText**

Das Grundkonzept von SchemaText beruht darauf, daß linguistisches Wissen über Textproduktionsprozesse berücksichtigt wird und die Anlage des Programms direkt beeinflußt.

Dabei wurden schon in der Konzeption des Programms den oben geschilderten Problemen der Technischen Dokumentation Rechnung getragen. Um die Archivierbarkeit der erzeugten Daten zu gewährleisten, wurde als internes Textformat HTML gewählt.<sup>4</sup> Sämtliche Textdaten eines Webbaumes (also neben den Inhalten auch Layoutinformationen, externe Dateireferenzen und die Verweisstrukturen) werden in einer gegliederten ASCII-Datei gespeichert, so daß die Daten durch einen Parser in jedes andere strukturierte Textformat überführt werden können, sollte dies durch die Entwicklung des HTML oder durch zukünftige Programmversionen einmal nötig werden.

Sowohl das Programm SchemaText als auch die programmierteile zur Texterstellung sind in Scheme programmiert – einem kleinen, aber sehr mächtigen LISP-Dialekt, der in der KI-Forschung häufig Anwendung findet. Für die Erstellung und Programmierung von SchemaText war Scheme auch deshalb besonders geeignet, da es in einer festgefügtten Norm vorliegt (Clinger/Rees 1991).

Insgesamt ist somit durch die Verwendung standardisierter, weit verbreiteter Basisformate eine hohe Archivierbarkeit der Texte gegeben und dem „Verfall“ der produzierten Texte weitgehend Einhalt geboten.

---

4 Die Standardisierung und Weiterentwicklung wird vom W3-Konsortium durchgeführt. Ziel ist es dabei, die Abwärtskompatibilität und Einheitlichkeit der Sprache zu gewährleisten (<http://www.w3c.org>, zuletzt eingesehen am 15.10.1998).

### 3.2 Textstruktur und Textproduktion

Nach van Dijk (1980) lassen sich in textueller Hinsicht drei verschiedene Ebenen unterscheiden: Die Mikrostruktur, bestehend aus Propositions- und Sequenzstrukturen, die durch die Anwendung von Makroregeln die Makrostruktur eines Textes aufbaut. Diesen beiden Strukturen als abstraktes Modell übergeordnet ist die Superstruktur, die „den Typ eines Textes“ (van Dijk 1980, 128) repräsentiert.

Die Superstruktur könnte somit als das psychologisch-strukturelle Modell einer Textsorte verstanden werden. Dabei muß sich jedoch dieses Modell nicht mit den alltagssprachlich bezeichneten Textsorten decken, sondern es kann im Detaillierungsgrad als auch im Referenzumfang von den „natürlichen“ Textsorten abweichen. Für die Zwecke dieses Aufsatzes bleibt jedoch als Essenz festzuhalten, daß die Superstruktur ein Modell beschreibt, mit dem unterschiedliche Texte in ihrer Struktur zusammengefaßt werden.

In der Technischen Dokumentation versucht man nun – mehr oder weniger intuitiv – sich diese Strukturähnlichkeit verschiedener Textsorten zunutze zu machen. Linguistisch gesehen ist z. B. das Erstellen einer DTD (Document Type Definition) in SGML<sup>5</sup> mit dem Versuch gleichzusetzen, Superstrukturregeln für eine Klasse von Texten zu formulieren. Problematisch ist in diesem Fall – und dies läßt sich in der Praxis immer wieder bei SGML-Projekten beobachten –, daß Textsorten und damit in einem gewissen Sinn auch Superstrukturen als prototypisch organisierte Felder zu verstehen sind (Heinemann/Viehweger 1991, 172ff.). Im Berufsalltag hat dies zur Konsequenz, daß zur Abdeckung weniger prototypischer Textexemplare entweder aufwendige Ausnahmeregeln definiert werden müssen oder daß auf bestimmte, sinnvolle Textstrukturen verzichtet werden muß, da diese nicht einem einmal festgelegten Prototyp entsprechen. DTDs tendieren deshalb dazu, im Laufe der Zeit immer unüberschaubarere Strukturen anzunehmen oder die Fortentwicklung eines Textkonzepts zu hemmen.

Bei SchemaText wurden deshalb die gegensätzlichen Bedürfnisse der Normierung und Flexibilität in anderer Weise berücksichtigt. Auf einer grafischen Oberfläche sind beim Starten des Programms zunächst zwei Fenster zu sehen, der sog. Schema-Editor und der Struktur-Editor (s. Abb. 1). Der Schema-Editor entspricht dabei in etwa der Textorganisation auf der Ebene der Superstruktur. Er ist somit eine abstrakte Schablone, die für eine Vielzahl ähnlich strukturierter Texte verwendet werden kann.

---

5 Eine gut verständliche Einführung in SGML bietet Szillat (1995).

Der Struktur-Editor dagegen stellt die konkrete Realisierung einer Makrostruktur dar. Somit werden sowohl für die Makrostruktur als auch für die Superstruktur Zusammenhänge zwischen den Textteilen verdeutlicht und damit insbesondere schreibunerfahrene (Hypertext)-Autoren unterstützt.

Wie in Abb. 1 zu sehen, werden die einzelnen Textteile durch Knoten (rechteckige Blöcke) und Unterknoten dargestellt; die Beziehungen der Textteile untereinander durch Links (Pfeile). Jedes dieser Elemente ist einzeln manipulierbar und programmierbar. Anders als bei SGML sind somit beide übergeordneten Textebenen individuell programmierbar, bis hin zu Ausnahmeregeln für einzelne Unterknoten.

Gleichzeitig ist es durch das grafikorientierte Anlegen von leeren Textknoten auch für einen einzigen Autor möglich, schnell Webbäume mit mehreren 1000 Dateien zu erstellen und zu pflegen (*Authoring in the Large*). Natürlich sind diese Dateien zu diesem Zeitpunkt nur mit automatisch generierten Texten gefüllt; dennoch ist dieses Verfahren von Nutzen, da so schon im Vorfeld die Stimmigkeit der Verlinkung und der Informationsstrukturierung geprüft werden kann.

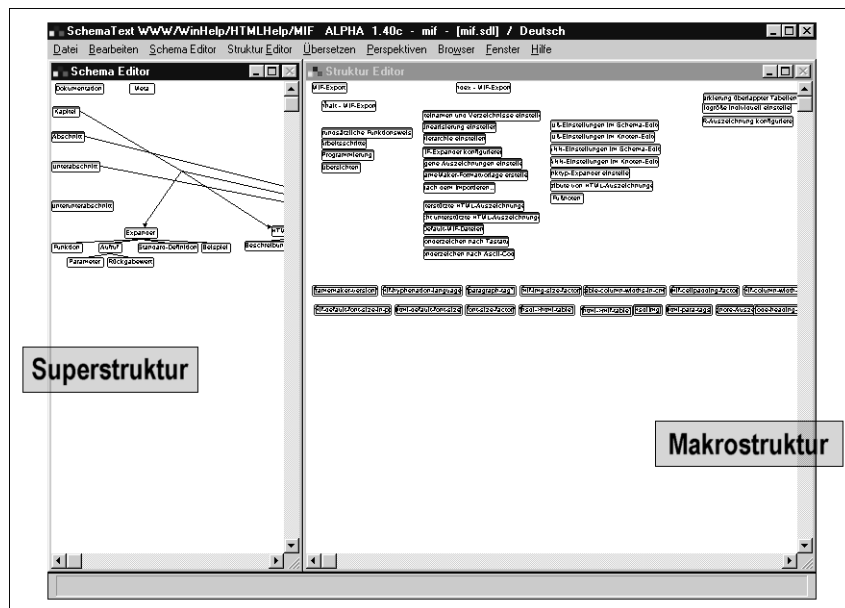


Abb. 1: Aufteilung des Arbeitsplatzes nach Super- und Makrostruktur

Die Mikrostruktur wird im Gegensatz zu Super- und Makrostruktur nicht durch die grafische Oberfläche bearbeitet, sondern durch einen herkömmlichen HTML-Editor<sup>6</sup>, den sog. Knoteneditor (s. Abb. 2).

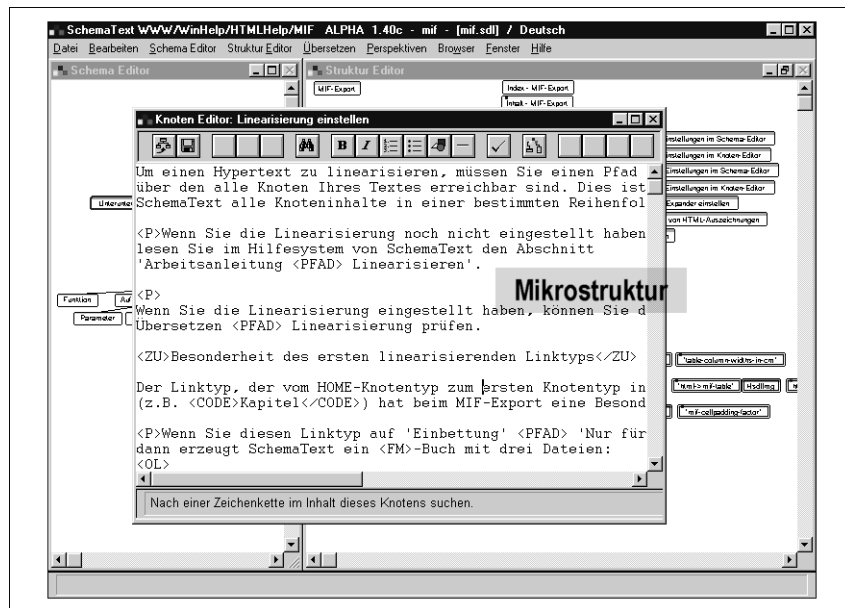


Abb. 2: Der Knoteneditor als Ebene der Mikrostruktur

Durch die Unterscheidung nach den drei verschiedenen Textebenen sind den Autoren insbesondere durch die Veränderung der Superstruktur umfassende Überarbeitungsmöglichkeiten an die Hand gegeben. Gleichzeitig läßt sich aber auch die Ebene der Superstruktur dazu nutzen, eine Norm für bestimmte Textklassen bereitzustellen, ohne im Detail so eingeschränkt zu sein, wie das üblicherweise bei SGML-basierten Lösungen der Fall ist. Als zusätzlicher Nutzen ist festzuhalten, daß SchemaText die abstrakte Textstruktur veranschaulicht und somit dem Autor einen kognitiven Zugang zu dieser Textstruktur bereitstellt. Damit

6 Der eingebaute HTML-Editor stellt jedoch grafische Menüs für die häufigsten HTML-Tags zur Verfügung. Eine andere Möglichkeit, die Mikrostruktur zu erstellen, besteht darin, die Texte in WinWord zu verfassen und über eine eingebaute Schnittstelle an SchemaText direkt zu übergeben.

---

schaftt das Programm eine motivationale Basis für eine Normierungsebene, die sonst im allgemeinen nur widerwillig eingehalten wird.<sup>7</sup>

### 3.3 Formate

Wie in Abbildung 2 zu sehen, wird die interne Textbasis bei SchemaText in HTML realisiert. Dies bedeutet jedoch nicht, daß die Ausgabeformate darauf beschränkt sind.

Neben HTML können auch andere Hypertextformate wie WinHelp und HTML-Help erzeugt werden. Auch lineare Textformate sind möglich: SchemaText hat einen Ausgabemodus für das FrameMaker Interchange Format (MIF) und für das Textsatzprogramm LaTeX.

Der HTML-Export ist dabei nur von den im Knoten-Editor eingesetzten Auszeichnungen (Abb. 2) beschränkt; dementsprechend sind derzeit alle Tags der Version 4.0 sowie Frames und CSS möglich.

Zusätzlich bietet SchemaText die Möglichkeit, sich eigene Tags zu definieren, die als Textbausteine dienen. So kann z. B. ein Tag für die Firmenanschrift definiert werden, der sich leicht „updaten“ läßt, auch semantische Auszeichnungen für Textpassagen lassen sich festlegen (z. B. Tip, Warnung, Beispiel), die dann einer normierten Form dargestellt werden. Beim Export werden diese individuellen Tags wieder in die regulären Formate zurückgeführt.

### 3.4 Perspektiven

Häufig besteht in der Technischen Dokumentation der Wunsch, verschiedene Varianten desselben Textes vorrätig zu halten. SchemaText erlaubt dem Autor, verschiedene „Perspektiven“, d. h. Texte mit identischer Ausgangsstrukturierung, nebeneinander in einer Datei zu halten. Strukturveränderungen können in jeder Perspektive auch weiterhin vorgenommen und auf einfache Weise an alle Perspektiven durchgereicht werden.

Dadurch wird z. B. die Arbeit des Übersetzers erleichtert. Er muß sich nun nicht mehr um den Aufbau eines Hypertext-Grundgerüsts kümmern, sondern kann seine Arbeit den Inhalten widmen. Andererseits steht es ihm auch offen, falls nötig von der Ausgangsperspektive abzuweichen und auf einfache Weise die Struktur seiner Texte an die jeweiligen sprachspezifischen Makrostrukturen anzupassen.

---

<sup>7</sup> Jedem, der schon einmal einen Tagungsband herausgegeben hat, ist wohl der Aufwand bekannt, der betrieben werden muß, damit sich die Autoren an die jeweiligen Stilrichtlinien halten.

Die Perspektivenfunktion von SchemaText kann jedoch auch zu anderen Zwecken genutzt werden. So existiert hier z. B. eine komfortable Möglichkeit, verschiedene Versionen eines Webbaums (beispielsweise eine reine Textversion neben einer grafischen und einer Frameversion) gleichzeitig zu pflegen und ihre Konsistenz zu sichern. Strukturänderungen können über einen Menüeintrag leicht abgeglichen werden.

### 3.5 Linearisierung

Ebenfalls eine häufige Anforderung in der Praxis ist es, aus einer Quelle Print- und Onlineversionen eines Textes zu erzeugen. In SchemaText lassen sich die einzelnen Teile der Makrostruktur linearisieren<sup>8</sup>; die Links werden dabei entweder durch die lineare Textstruktur wiedergegeben oder in Querverweise und Fußnoten aufgelöst. Entgegen der sonst üblichen Lösung im Cross-Media-Publishing, lineare Texte zu hypertextualisieren (Storrer 1997, 124), wird hier der umgekehrte Weg angestrebt, nämlich eine Komplexitätsreduktion multidimensionaler Textstrukturen unter Zuhilfenahme des Textsortenwissens der Autoren zu erreichen. Die so erzeugten linearen Texte konnten in einem Test von den Versuchspersonen nicht von linear geschriebenen Texten unterschieden werden – eine Anforderung, die sich im umgekehrten Fall (hypertextualisierte Lineartexte) nur selten bzw. nur mit erheblichem Nachbearbeitungsbedarf einlösen läßt (Storrer 1997, 137f.).

## 4 Probleme

Bisher war zu einem großen Teil nur von den Vorzügen des Programms die Rede. Doch sollen auch die Nachteile der Anwendung hier nicht unter den Tisch fallen. Sie gründen – wie so häufig in der Technischen Dokumentation – darin, daß zur Erstellung guter Dokumentation neben dem Sachwissen i. e. S. auch Sprachwissen und Computerkenntnisse nötig sind.

### 4.1 Textstrukturierungsfähigkeit

Während die „Füllung“ eines bestehenden Schemas mit Inhalten eine vergleichsweise einfache Tätigkeit ist, stellt die Erstellung eines Schemas komplexe Anforderungen an die Textstrukturierungsfähigkeit des Autors. Vielen Autoren, insbesondere in der Technischen Redaktion, gelingt es nicht, sich von der Ebene der Mikro- und Makrostruktur zu lösen, so daß die „Superstruktur“ unnötig komplex angelegt wird.

---

8 Zum Begriff der (Ent)Linearisierung siehe Freisler (1994, 22ff.); Nickl (1996, 392); Nickl (1997a, 99f.).



Aus diesem Grund ist es meist nötig, zumindest zu Beginn der Arbeit den Entwurf einer Superstruktur einem Textspezialisten (im allgemeinen einem Linguisten) zu überlassen oder auf die weniger spezifischen Beispielschemata zurückzugreifen, die mit dem Programm mitgeliefert werden. Mit der Zeit wird aber, insbesondere auch durch die Visualisierung der Superstruktur, die das Programm bietet, Textsortenwissen in diesem Bereich aufgebaut, so daß mit zunehmender Erfahrung ein Lerneffekt eintritt und auch ein selbständiges Arbeiten möglich wird.

#### 4.2 Programmierfähigkeit

Die Eigenschaften der einzelnen Knoten und Links werden durch sog. Expander bestimmt, d. h. durch mehr oder weniger kurze Scheme-Programme. Problematisch ist hier nicht nur, daß Schemekenntnisse im allgemeinen weniger verbreitet sind als z. B. VisualBasic oder C++-Kenntnisse. Zusätzlich wird die Programmierung nämlich noch durch die Tatsache erschwert, daß für Scheme keine komfortable Programmierumgebung vorliegt.

Dies bedeutet, daß ein komplettes Schema nur dann von einem einzelnen Autor erstellt werden kann, wenn er linguistisches und programmiererisches Wissen in sich vereint, ein Fall der angesichts der Seltenheit von Schemeprogrammierern nicht oft vorkommen wird.

#### Literatur

- Clinger, William/ Rees, Jonathan (eds.) (1991): Scheme. Revised 4th Report on the Algorithmic Language Scheme. In: ACM Lisp Pointers 3 (4)
- Freisler, Stefan (1994): Hypertext – eine Begriffsbestimmung. In: Deutsche Sprache. Zeitschrift für Theorie, Praxis, Dokumentation 1 (22), 19-50
- Heinemann, Wolfgang/ Viehweger, Dieter (1991): Textlinguistik. Eine Einführung. Tübingen: Niemeyer [RGL; 115]
- Knorr, Dagmar (1997): Verwaltung von Fachtextinformationen. Anforderungen an Nutzer und Hilfesysteme. In: Knorr, Dagmar/ Jakobs, Eva Maria (Hrsg.): Textproduktion in elektronischen Umgebungen. Frankfurt/M. u. a.: Lang [Textproduktion und Medium; 2], 67-86
- Nickl, Markus (1996), Web Sites. Die Entstehung neuer Textstrukturen. In: Bollmann, Stefan/ Heibach, Christiane (Hrsg.): Kursbuch Internet. Anschlüsse an Wirtschaft und Politik, Wissenschaft und Kultur, Mannheim: Bollmann, 389-400
- Nickl, Markus (1997a): Textstrukturen im WWW – linguistische Analyse und praktische Bewertung. In: Eickemeyer, Klaus/ Adelhard, Hermann/ Artmann, Hans/ Habermaier, Mechthild/ Krieger, Michael/ Rust, Michael/ Strucken, Berthold (Hrsg.): Technische Information in elektronischen Medien. Der aktuelle Stand der Diskussion, zusammengestellt anlässlich der Fachtagung T.I.E.M. '97, Lübeck: Schmidt-Römhild, 95-105
- Nickl, Markus (1997b): Norm und Normierung in der Technischen Dokumentation. In: Mattheier, Klaus (Hrsg.): Norm und Variation. Frankfurt/M. u. a.: Lang, 137-150
- Storrer, Angelika (1997): Vom Text zum Hypertext. Die Produktion von Hypertexten auf der Basis traditioneller wissenschaftlicher Texte. In: Knorr, Dagmar/ Jakobs, Eva Maria (Hrsg.): Textproduktion in elektronischen Umgebungen. Frankfurt/M. u. a.: Lang [Textproduktion und Medium; 2], 121-139

- 
- Szillat, Horst (1995): SGML. Eine praktische Einführung. Bonn u.a.: International Thomson Publishing
- Van Dijk, Teun A. (1980): Textwissenschaft. Eine interdisziplinäre Einführung. Tübingen: Niemeyer